2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Placeit! A Framework for Learning Robot Object Placement Skills

Amina Ferrad*[1], Johann Huber*[1], François Hélénon[1],
Julien Gleyze[1], Mahdi Khoramshahi[1], and Stéphane Doncieux[1]

***Abstract*— Robotics research has made significant strides in learning, yet mastering basic skills like object placement remains a fundamental challenge. A key bottleneck is the acquisition of large-scale, high-quality data, which is often a manual and laborious process. Inspired by *Graspit!*, a foundational work that used simulation to automatically generate dexterous grasp poses, we introduce *Placeit!*, an evolutionary-computation framework for generating valid placement positions for rigid objects. Placeit! is highly versatile, supporting tasks from placing objects on tables to stacking and inserting them. Our experiments show that by leveraging quality-diversity optimization, Placeit! significantly outperforms state-of-the-art methods across all scenarios for generating diverse valid poses. A pick&place pipeline built on our framework achieved a 90% success rate over 120 real-world deployments. This work positions Placeit! as a powerful tool for open-environment pick-and-place tasks and as a valuable engine for generating the data needed to train simulation-based foundation models in robotics.**

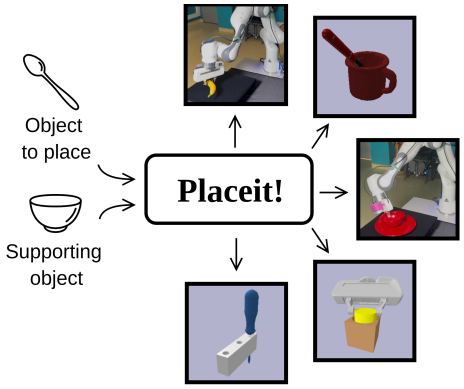

Fig. 1. *Placeit!* takes two object meshes as input and automatically finds diverse ways for them to interact. The resulting poses can be applied to a wide variety of scenarios, enabling robust robotic manipulations.

I. INTRODUCTION

Despite decades of effort in robot learning, basic manipulation skills like grasping and placing objects remain a significant challenge [1]–[5]. As the field has evolved, data acquisition has grown to be a primary consideration for the community [3], [6]–[8], [26]. This is mostly due to the prevailing belief that major breakthroughs, which one could refer to as the "ChatGPT moment" of robotics, will arise from the availability of large-scale data [10].

This has led to two primary data acquisition paradigms. The first involves collecting massive amounts of high-quality data from the physical world [2], [11]–[14]. This approach provides realistic information but the data collection process is slow and hardly scalable. The second paradigm utilizes simulation to accelerate data collection and safely train robot control models [1], [15], [16]. Recent research often combines both, either to build foundational models [17] or to enhance simulation fidelity through methods like real2sim2real [18].

Simulators should allow to automatically generate data for robotic learning, and scale-up the process in a safer and more efficient manner than in the physical world. However, a considerable amount of work also requires manually collected data in simulation [17], [19]: the automatic synthesis of meaningful training data remains a significant challenge. Although there has been extensive research on automated grasping [6], [7], [20], other manipulation tasks, such as placing, have not been as thoroughly investigated.

The automatic generation of data for robotics was strongly inspired by the seminal work of *Graspit!* [21], a framework that generates a set of dexterous grasps in simulation based on analytic criteria, given the 3D models of an object and a gripper. Inspired by this foundational work, we introduce *Placeit!*, an evolutionary-computation based framework for automatically generating valid placement positions of rigid objects in a simulated environment.

This framework is versatile, taking two objects as input—one to be placed and a support object. This allows it to handle various placement scenarios, including table-top, hanging, stacking, and inserting (Fig. 1). Our approach leverages Quality-Diversity (QD) optimization, a family of evolutionary algorithms that has proven effective for generating diverse and high-quality data for robot learning [22].

The key contributions of this work are the following:

- We introduce Placeit!, a versatile framework for automatically generating diverse and valid placement positions of rigid objects in various simulated scenes;
- We demonstrate that a state-of-the-art QD method leveraged in the Placeit! framework outperforms standard approaches in most of the considered scenarios by a large margin;
- We also introduce *Quality-Diversity Grasp-and-Place* (QDGP), a pipeline built on the top of Placeit! to do pick&place tasks in various scenarios. QDGP achieved about 90% success rate over 120 deployments through the use of domain randomization to sample the most promising poses.

* equal contribution and corresponding authors
[1]Sorbonne Université, CNRS, Institut des Systèmes Intelligents et de Robotique, ISIR, F-75005 Paris, France {ferrad, huber, helenon, gleyze, khoramshahi, doncieux}@isir.upmc.fr

The code and models will be made publicly available upon acceptance of this article. Visualizations of both the simulated and the real-world experiments are provided with the attached video.

II. RELATED WORKS

Learning to Place in Robotics. Research in robot placement has long relied on manually labeled or expert-provided data. This approach involves defining placement poses programmatically for predefined objects, which may include primitive shapes [23] or common household items [24]–[26]. Numerous works that address object placement, including those leveraging supervised learning from human demonstrations, fall into this category. Examples range from studies focused exclusively on placing [27] to those with a broader scope covering various manipulation tasks [2], [17]. Another common approach is the use of Reinforcement Learning (RL), which often requires carefully hand-crafted reward functions to guide the learning process [42] or constrains the problem to a specific, more tractable setup [28]. Many pick-and-place scenarios simplify the placement problem by merely defining a target position for the end-effector, typically above a large container, before releasing the object [4], [5], [15], [29]. Existing methods are heavily reliant on human intervention, making data acquisition slow, tedious, and difficult to scale. This manual dependency is a significant bottleneck for generating the large-scale datasets required for modern foundation models in robotics. Our framework overcomes this limitation by automatically generating valid placement poses across diverse scenarios.

Many works on robot placement rely on analytical criteria to sample valid poses. A common approach involves aligning vertices or mesh faces of an object with a target surface in simulation. A candidate pose is typically considered valid if the projection of its center of mass onto the surface, along the gravity vector, falls within the surface's polygon [30]–[32]. However, this method has significant limitations. It is highly sensitive to the quality of the object's mesh, overlooks the dynamics of physical interactions during placement, and fails to differentiate between stable and unstable placement positions, as long as the geometric criteria are met. In contrast, our framework explores the physical interactions between the object and the support in a simulated environment. This enables us to not only validate a placement but also to quantify its stability. Furthermore, we use a domain-randomization-based approach to simulate local perturbations on the placed object, which helps in identifying stable poses versus those that are local maxima of potential energy.

Recent research has increasingly used simulations to test the physical interactions between objects and surfaces for generating placement poses. Zhao et al. [3] leveraged principal component analysis to sample placement poses by aligning objects with their supports. Noh et al. uniformly sampled object orientations and simulated dropping them onto a plane to discover valid placement poses [8]. Simeonov et al. [26] rely on manually coded procedures to sample initial candidates, which are then validated through simulation. While these works have made contributions to specific areas—such as automatic labeling for tabletop scenarios [8], object alignment for stacking and inserting [3], or addressing diverse placement scenarios with human-labeled poses [26], our framework is designed to automatically sample placements across all of these scenarios. Moreover, our sampling scheme, which is based on quality-diversity optimization, significantly outperforms the automatic sampling methods used in the aforementioned papers across nearly all tested scenarios.

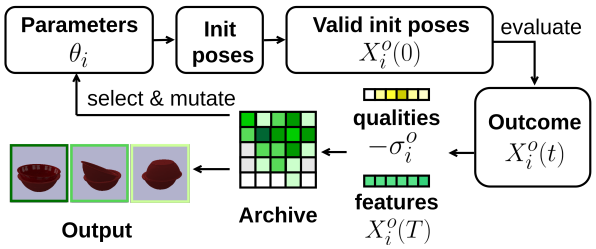

Fig. 2. **Placeit! principle.** The framework uses QD optimization to explore the interactions between an object and a support. A mutation-selection process efficiently explores the space of possible placement solutions from a parameter space.

Quality Diversity. Quality-Diversity methods are optimization algorithms that aim to generate a set of diverse and high-performing solutions to a given problem. An increasing amount of works are dedicated to using QD for generating data for robot learning. This includes the generation of demonstrations for locomotion [33], grasping trajectories [6], or object-centric grasps [34], but also the production of adversarial objects to grasp [35]. Application of QD algorithms for the acquisition of manipulation skills are however limited to grasping [6]. The potential of these algorithms to generate diverse data in hard exploration problems suggests that it could be leveraged for prehensile manipulation tasks. The present paper fills this gap by investigating how to leverage QD optimization to build a versatile tool for generating placement position of objects.

III. METHOD

A. Placing

Let $P \subseteq \mathrm{SE}(3)$ be the space of rigid object poses, and $P^* \subseteq P$ the space of stable poses. This section discusses how the literature generates diverse, stable object placements for robotic manipulation—that is, exploring a *parameter space* Θ to find associated stable positions in P^*.

Definition. We consider the placement of an object o in interaction with a supporting object s. For an individual θ_i, let $X_i^o(t) \in P$ be the pose of a rigid body o at time t in the supporting object frame that we assume static. The object o is placed on s if and only if for each $t \in t_{bvp}, ..., t_{evp}$:

$$\begin{cases} \sum \vec{F}_{ext}(t) = \vec{0} \\ \sum \vec{\tau}_{ext}(t) = \vec{0} \\ in\_contact_{o,s}(t) = 1 \end{cases} \quad (1)$$

where $\vec{F}_{ext}$ and $\vec{\tau}_{ext}$ are respectively the external forces and external torques applied on o at timestep t, $t_{bvp}, t_{evp} \in \mathbb{N}^+$ are respectively the timestep that begins the validation and the one that ends the validation of the placement (with $t_{bvp} < t_{evp}$), and $in\_contact_{o,s} : \mathbb{N}^+ \rightarrow \{0,1\}$ is a binary function that returns 1 if o and s are in contact at timestep t and 0 otherwise. From an empirical perspective, the placement of o on s can be empirically verified if for each $t \in t_{bvp}, ..., t_{evp}$:

$$\begin{cases} \vec{v}_{o/s}(t) \approx \vec{0} \\ in\_contact_{o,s}(t) = 1 \end{cases} \tag{2}$$

where $\vec{v}_{o/s}$ is the velocity (linear and angular) of o in the frame of s.

Sampling schemes. To discover placing poses, previous works essentially relied on hand-crafted search space to guide the search toward promising solutions, and then explore P by randomly sampling Θ.

The existing literature on sampling placement poses primarily uses three main strategies. The first involves aligning two surfaces of the considered objects [30]–[32]. The second aligns objects based on their geometry, which is particularly effective for problems with symmetrical properties, such as peg-in-hole or stacking [3]. The third strategy samples random poses for one object relative to another, such as placing an object above a table with a random orientation, to find diverse placement poses [8].

These methods often rely on a predefined search space combined with a brute-force search. Placeit!, moves beyond these limitations by integrating the search space with an optimized exploration strategy based on recent advances in evolutionary computation.

B. Placeit!

Definition. Let $\theta \in \Theta$ be an *individual*. Let $\Phi \subseteq \mathbb{R}^{n_b}$ be the *feature space*, and $\zeta_\Phi : \Theta \rightarrow \Phi$ the *feature function*, which assigns a *feature descriptor* $\phi_\theta = \zeta_\Phi(\theta)$ to each θ. The *fitness function* is $f : \Theta \rightarrow \mathbb{R}$, and $d_\Phi : \Phi^2 \rightarrow \mathbb{R}$ is a distance function within Φ. The goal is to generate an *archive* A such that:

$$\begin{cases} \forall \phi \in \Phi_{reach},\ \exists \theta \in A,\ d_\Phi(\zeta_\Phi(\theta), \phi) < \epsilon \\ \forall \theta' \in A,\ \theta' = \text{argmax}_{\theta \in N(\phi_{\theta'})} f(\theta) \end{cases} \tag{3}$$

where $\Phi_{reach} \subseteq \Phi$ is the *reachable feature space*, $\epsilon \in \mathbb{R}^{+*}$ defines the density of Φ_{reach} paving, and $N(\phi_{\theta'}) = \{\theta \mid neighbor_{d_\Phi}(\phi_\theta, \phi_{\theta'})\}$ is the set of solutions with close projections in Φ. ζ_Φ is deterministic.

Principle. The core principle of Placeit! is to use quality-diversity optimization to discover a diverse set of valid and stable placement poses (Fig. 2). The framework leverages the powerful exploration properties of evolutionary algorithms [6], [22], making the discovery of these poses both automatic and sample-efficient. Unlike other methods that rely on priors specific to certain objects and scenarios [3], [8], [30]–[32], Placeit! is a generic framework that can be applied to any 3D objects provided as input for both the object to be placed and the supporting object.

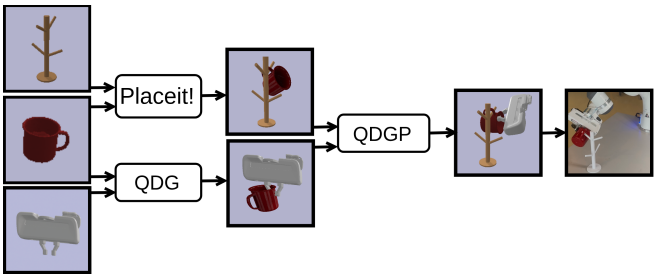

Fig. 3. **QDGP pipeline.** Grasp poses are generated using QDG [34], and placing poses with Placeit!. The combined poses are validated before being deployed on the physical robot to perform the full pick-and-place sequence.

Evaluation. The process begins by randomly sampling a population of parameters $\{\theta_i\}_{i=0}^n$ from the parameter space Θ. These parameters are then used to generate a set of initial poses $\{X_i^o(0)\}_{i=0}^n$ for the object o relative to the supporting object s. We discard any poses that result in an overlap between the two objects.

Each valid initial pose is then evaluated in a simulation. Gravity is applied to the object o for a duration of T time steps, allowing it to fall and settle. The result of this simulation is a sequence of poses $\{X_i^o(t)\}_{i=0}^n$ for $t \in 0, ..., T$. A placement attempt is considered successful if the object reaches a stable state at the end of the simulation. This stability is determined by the variance of the object's pose over the final $\delta_t = t_{evp} - t_{bvp}$ steps of the simulation (here $t_{evp} = T$). Specifically, a placement is successful if:

$$\sigma_i^o < \sigma^{st} \mid \sigma_i^o = \sigma(\{X_i^o(t)\}_{t=T-\delta_t}^T) \tag{4}$$

where σ_i^o is the variance of the object's pose, and σ^{st} is a predefined threshold that distinguishes stable poses from unstable ones.

For each successful placement, a quality score (fitness) and a descriptor (feature) are computed. The fitness of θ_i is defined as $f_{\theta_i} = -\sigma_i^o$, such that a lower variance (higher stability) corresponds to a better fitness. The feature is the final pose of the object relative to the support, $\phi_{\theta_i} = X_i^o(T)$, which describes the outcome of the placement.

Optimization. The core of the Placeit! framework is its use of a QD algorithm, which maintains an archive of solutions [36]. This archive acts as a memory of the exploration process, storing the highest-performing solution for each distinct feature. Solutions, defined by their initial poses and corresponding outcomes, are added to the archive based on their feature and fitness. If a cell in the archive (corresponding to a specific feature range) is empty, a new solution is added. Otherwise, if a solution with a higher fitness is found, it replaces the existing one. This process progressively populates the archive with a diverse set of high-quality solutions.

New parameters for the next iteration are sampled from this archive and mutated with random noise, allowing the algorithm to build on previously successful discoveries. The algorithm's output is the optimized archive, which contains a collection of diverse and robust final poses. The objective is

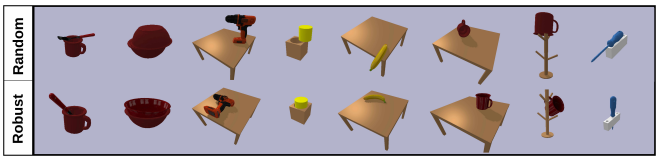

Fig. 4. **Example solutions.** The top row shows solutions randomly sampled from a CMA_MAE run. The bottom row shows robust solutions from the same run, identified using our domain randomization criterion.

to optimize the coverage of the *successful feature space* $\Phi^s \subseteq \Phi$ which represents all final poses that meet the stability criteria. The QD algorithm's local competition ensures that the optimization focuses on finding the most stable poses within each feature region.

The final output of the algorithm is the archive of robust, diverse final poses $X_i^o(T)$. These poses can then be used for deployment on a physical robot. While some complex cases may require dedicated methods for the approach phase [37], this separate problem is outside the scope of this work.

Domain randomization. The fitness function we use efficiently identifies valid placement states in a single evaluation, a common approach in QD optimization [22]. However, a rigid body's true stability is determined by its resistance to small perturbations, which relates to potential energy analysis [38]. In this work, we focus on identifying stable equilibria in the sense of a pose where the object remains at rest without toppling or rolling away under small perturbations. To achieve this, we introduce a domain randomization-based filter that distinguishes stable equilibria (minima of potential energy) from unstable ones (maxima).

For a given final object state, $X_i^o(T)$, we apply a perturbation $\epsilon \in \mathbb{R}^6$ to get a new state $\tilde{X}_i^o(T) = X_i^o(T) + \epsilon$. We then evaluate $\tilde{X}_i^o(T)$ by applying gravity for $\delta_{t_{DR}}$ additional steps, which gives us a set of states, $\{\tilde{X}_i^o(T+t)\}_{t=0}^{\delta_{t_{DR}}}$. A placing pose, $X_i^o(T)$, is considered stable if:

$$\sigma_i^{o^{DR}} < \sigma^{DR} \mid \sigma_i^{o^{DR}} = \sigma(\{\tilde{X}_i^o(T+t)\}_{t=0}^{\delta_{t_{DR}}}) \qquad (5)$$

Here, $\sigma_i^{o^{DR}}$ is the variance of the object's state during the $\delta_{t_{DR}}$ stabilization steps after the perturbation, and σ^{DR} is a predefined variance threshold. Defining σ^{DR} is challenging since it includes both positional and rotational variance. The value of σ^{DR} is here defined empirically, making the "robust" label subjective. However, this evaluation method still allows us to determine which states are more stable than others. Our empirical results show that this is a powerful analytical tool (section V).

To properly assess an object's potential energy, its resistance to a range of small perturbations must be tested, as a single perturbation might not reveal an unstable state. We perform this test $M \in \mathbb{N}^{*+}$ times. A larger value of M gives a better estimate of the potential energy but requires more computation.

Quality-Diversity Grasp-and-Place. To explore the usage of Placeit! for pick&place tasks, we introduce QDGP, a pick-and-place pipeline based on quality-diversity generated data (Fig. 3). First, a set of placing poses $X_i^o(T)$ is generated for a given scenario. Simultaneously, a set of grasping poses is produced on the object o and the gripper using QD optimization [34]. For each placement pose, the generated grasps are evaluated in a simulation to guarantee their validity within the placing setup. This process yields a set of grasps, corresponding placement poses, and associated quality metrics—including whether $X_i^o(T)$ is a stable equilibrium with respect to σ^{DR}. The object is then grasped and placed using the corresponding poses and a motion planner.

IV. EXPERIMENTS

This section presents the essential information about our experiments. Further details, including hyperparameters, can be found in the Supplementary Materials.

Tasks. An overview of the tasks we considered can be found in Fig. 4. We defined 8 scenarios in simulation to compare the placement sampling schemes. These scenarios included: 3 tabletop tasks (*banana-on-table*, *mug-on-table*, *power drill-on-table*); 1 stacking task (*stack-bowl*); 3 insertion tasks (*peg-in-hole*, *spoon-in-mug*, *screwdriver-on-support*); and one hanging scenario (*hanging-mug*). These tasks were chosen to demonstrate the framework's versatility across various placement challenges, reflecting different levels of complexity. They also correspond to scenarios proposed in works used as baselines [3], [8], [26], [31].

Compared methods. Our experiments evaluate two critical components of the Placeit! framework: the search space and the optimization algorithm. We structured the comparison into four main parts: state-of-the-art baselines, QD approaches, search space baselines, and a final combination of a prior-based search space with QD.

The existing literature on generating placement poses primarily relies on three types of prior-based methods:

- *face_alignment*: This approach involves randomly sampling and aligning a triangle mesh from both the object to be placed and the support object [4], [31], [32]. A random orientation is applied, and the validity is checked by simulating gravity;
- *pca_alignment*: This method uses Principal Component Analysis (PCA) to align the two objects based on their symmetrical axes. A random perturbation is then applied, and the object is dropped to check its validity [3];
- *rand_sample*: This approach samples a random initial pose for the object relative to the support, then applies gravity to see if it settles into a stable state [8].

We also included several Quality-Diversity methods, which utilize a naive, prior-less search space where the object's position and orientation are directly controlled by the algorithm's genome. The object is then "dropped" to test its stability. The QD variants we included are:

- *ME_rand*: A standard MAP-Elites baseline [36];
- *ME_scs*: A success-greedy variant of MAP-Elites, known for its efficiency in tasks like grasping [6];
- *CMA_MAE*: A state-of-the-art QD approach that has performed well across many domains [39].

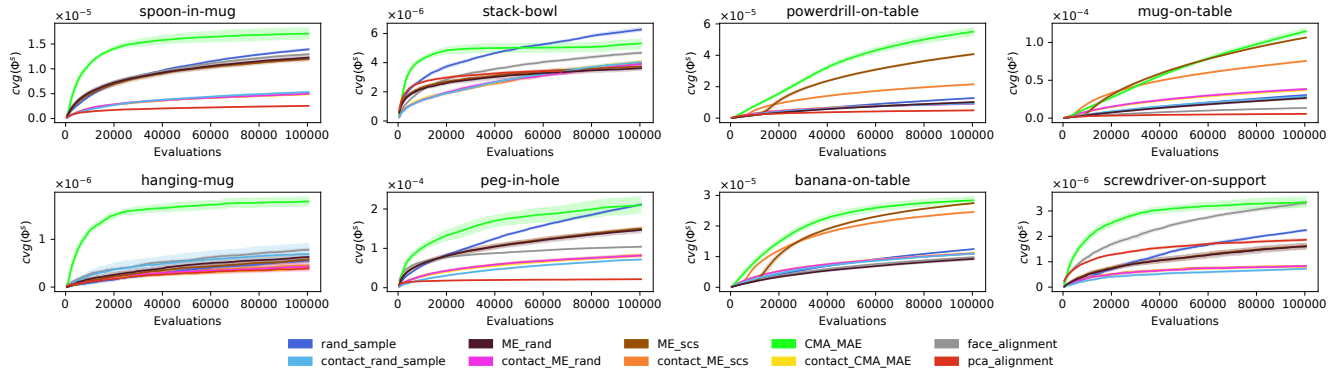

Fig. 5. **Comparison between methods.** This figure shows the evolution of the diversity of placement solutions found ($cvg(\Phi^s)$) throughout the optimization process for each simulation scenario. The CMA_MAE method without priors significantly outperforms all other methods across most scenarios.

To investigate the impact of reducing the search space, which has been shown to improve QD performance [34], we introduced a prior-based search space called *contact_rand_sample*. This approach randomly samples an initial state for the object relative to the support object's surface. A triangle mesh on the support object is selected, and the object to be placed is positioned and oriented within a cone around the triangle's normal—similarly to what Eppner et al. did for grasping [40]. We also created combinations of the prior-based search space with each of the QD methods (*contact_ME_rand*, *contact_ME_scs*, and *contact_CMA_MAE*) to evaluate their combined performance.

Pick&Place in the real-world. To evaluate the quality of the placement poses found in simulation and their applicability in the real world, we conducted a sim-to-real transfer experiment. A set of 7 diverse scenarios were chosen, including tabletop, insertion, stacking, and hanging tasks. The scenarios used in this transfer experiment differ slightly from those in our main simulated experiments due to the accessibility of corresponding physical objects (e.g., the stack-bowl scenario was replaced with a bowl-on-plate). We performed dedicated runs of Placeit! for these specific scenarios to generate the poses for real-world testing. A quality label based domain randomization (eq. 5) is also computed to distinguish robust poses from the ones at an unstable equilibrium.

Experiments were performed using a Franka Research 3 robotic arm equipped with its native two-finger gripper. The objects primarily came from the YCB dataset [41], but some were 3D-printed, such as a custom peg-in-hole task and a hanger for the mug to match a baseline study by Zhao et al. [3]. QDGP relies on MoveIt as a motion planner.

Evaluation metrics. The simulation experiments aim to find the best approach for generating a placing position in any of the considered scenarios. To do so, we measure the coverage of Φ^s, noted $cvg(\Phi^s)$, which is the ratio of successful placing states relative to all the theoretically possible placing states in Φ. To evaluate the quality of the generated placing poses, we deploy them in the real world and measure whether the attempt is successful or not. A placing position is considered successful if the object reaches a stable state in the real world that is similar to the one found in simulation. Therefore, a placing attempt that accidentally leads to another valid placing state in the real world is not considered a success.

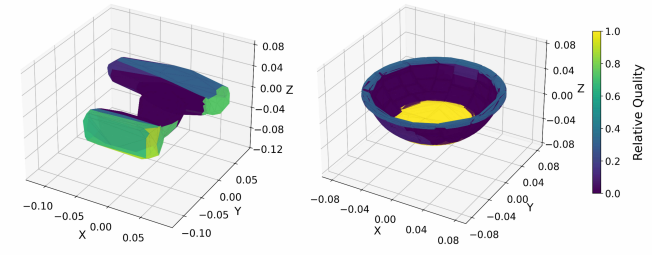

Fig. 6. **Placing heatmaps for 2 table-top scenarios.** The heatmaps show the relative quality of placement contacts. Areas of the mesh that most frequently contact the table have higher values (1 = always robust placement, maximum fitness; 0 = always fragile placement, minimum fitness). See supplementary materials for details.

V. RESULTS AND DISCUSSION

Placing in Simulation. As shown in Fig. 5, our experiment compares the performance of various methods. The QD state-of-the-art method without any prior (CMA_MAE) significantly outperforms all other approaches, particularly in the short run ($<$ 20k evaluations), where it excels across all scenarios. While random sampling (rand_sample) can eventually generate a greater diversity of poses in the long run (100k evaluations), much of this diversity is often redundant or fragile. CMA_MAE, by contrast, generates both diverse and high-performing solutions efficiently, avoiding less promising areas of the search space. This difference explains why random sampling can become competitive with CMA_MAE in specific long-run scenarios, such as rotations around the symmetrical axis in stack-bowl or peg-in-hole.

State-of-the-art methods relying on alignment priors (face_alignment, pca_alignment) are consistently outperformed by most of our QD variants. Although these priors can be effective in specific, symmetric scenarios (e.g., screwdriver-on-support, stack-bowl) or for insertion-like tasks (e.g., peg-in-hole, hanging-mug), their performance

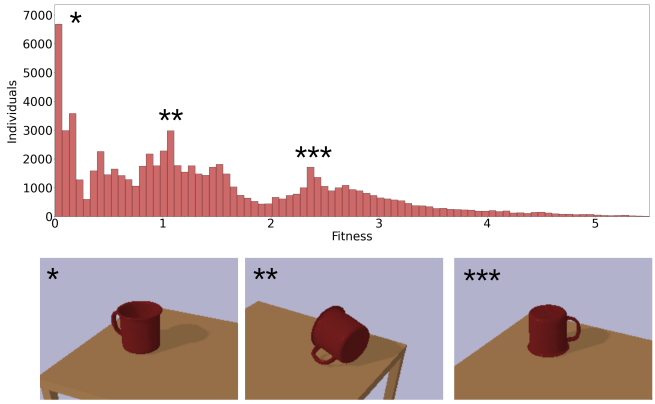

Fig. 7. **Resistance to local perturbations.** The histogram shows the distribution of placing poses based on their pose variance after a local perturbation (fitness). This data, computed from 10 seeds and averaged over 10 random perturbations per pose, reveals distinct modes corresponding to different stable ways for objects o and s to interact. The visualizations are randomly sampled poses from each of these peaks. This example highlights a limitation of the used simulator: placing the mug on its side (*) is much more stable than placing it on its flipped side (***).

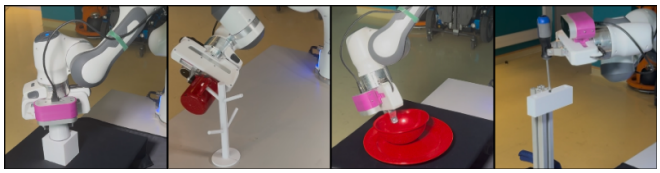

Fig. 8. **Real-world transfer.** The proposed framework produces diverse placement poses that achieve a high success rate when deployed in the physical world. This demonstrates the reliability of our approach for robotic manipulation learning.

is highly task-dependent. In contrast, our generalist QD approach (CMA_MAE) consistently provides superior sample efficiency, regardless of the scenario.

Our experiment also reveals that reducing the search space by forcing close contact between objects (contact_*) does not improve search efficiency for placing tasks. This finding contradicts its effectiveness in grasping scenarios [34]. We attribute this to the sparsity of the problem. Placing is a less sparse task than grasping, meaning a higher ratio of attempts yield successful solutions, which has a notable impact on QD methods' performance [6]. This also explains why CMA_MAE dominates ME_scs on most tasks. While ME_scs excels in problems with very low success rates and a concentrated region of interest [6], CMA_MAE performs best when it can rely on more frequent exploitable signals to balance exploration and exploitation [39]. This result highlights the need for a better understanding of the link between QD optimization algorithms and task sparsity, which is a promising direction for future work.

It is worth noting that the time per evaluation is comparable across methods achieving similar performance, as each evaluation constitutes the dominant computational cost. For the best-performing methods, the total runtime was approximately 5 to 30 minutes on a 40-core CPU machine.

Interaction properties. Since the best-performing methods in our study thoroughly explore how two objects interact, we can infer physical properties from the resulting set of object states. For instance, our framework can identify the parts of an object most likely to contact a support for a given scenario, as seen in Fig. 6. By creating a continuous map, we can pinpoint different placement modalities. When plotting the distribution of state variations after applying a local perturbation from a given placement pose (Fig. 7), the resulting modes correspond to the number of ways the objects can be placed on the support. Thus, the Placeit! framework can automatically identify how objects can be placed on one another, which allows for a broader analysis of object similarities and manipulation invariants. For example, all objects with a handle-like subpart could be automatically identified as something that can be hung in various ways—a property that would have emerged from prior interactions of similar objects with hangers.

This approach also reveals properties related to the simulation environment itself. For example, Fig. 7 shows that in Bullet, placing the YCB mug in its common-use pose (*) is more stable than flipping it (***). This is counterintuitive, as in the physical world, placing a mug on its top or bottom face typically yields comparable stability, both being less stable than on its side (**). In the simulation, the modeling of contacts between the 3D-scanned object and the surface causes the object state to vary significantly more when placed in pose (***) than in pose (*) or (**). This is another instance where domain randomization can stress the limits of simulations and be used as a tool for analyzing and improving both physics engines and rigid body meshes [20].

Scenarios	Random	Robust
banana-on-table	0.90	1.00
mug-on-table	1.00	1.00
peg-in-hole	0.56	0.78
spoon-in-mug	0.60	0.90
bowl-on-plate	0.90	1.00
screwdriver-on-support	0.40	0.60
hanging-mug	0.00	1.00
Total	0.62	0.90

TABLE I

SIM2REAL TRANSFER RESULTS.

Pick&Place in the real-world. Table I shows the successful sim-to-real transfer ratios achieved using QDGP across various placing scenarios. In every case, sampling poses that satisfy the stable equilibrium criterion leads to a higher transfer ratio compared to random sampling. While both methods perform well on easier scenarios like mug-on-table and bowl-on-plate, the difference is far more significant for tasks that require precision, such as peg-in-hole, screwdriver-on-support, and hanging-mug. This demonstrates that our domain randomization approach is effective at identifying which simulated placement poses are most likely to succeed in the real world by distinguishing local or global extrema of potential energy.

Fig. 8 shows several examples of successful deployments. The Placeit! approach led to a high success rate in plac-

ing manipulations for all scenarios, proving its utility for direct deployment in both industrial environments (e.g., screwdriver-on-support) and service robotics (e.g., bowl-on-plate). The demonstrated effectiveness of this approach in real-world scenarios confirms its potential for broader applications, such as learning to manipulate objects on-the-fly by scanning an environment [43] or generating labels for learning placing policies [26]. The framework's versatility, combined with the sample efficiency of QD optimization and the quality of its sim-to-real transfer criteria, makes it a promising tool for making future manipulation task learning easier and more efficient.

Limitations. This work focuses on the placement phase of object manipulation, assuming the preceding approach can be handled with standard planning. This assumption may not hold true in dense scenes [37]. We believe the data generated by our framework unlocks a critical part of the placing skill, and future work could explore how to integrate these placements with dedicated methods, such as reinforcement learning, to address this challenge. Furthermore, our experiments were limited to rigid, non-deformable objects. While the Placeit! framework is versatile enough to handle other types, extending our experimental results to include deformable or articulated objects is a promising direction for future research. Finally, Placeit!'s performance is highly dependent on the quality of its simulation environment, since it relies entirely on it. The success of our approach is critically tied to the quality of the object meshes, the accuracy of their inertial properties, and the precise modeling of contacts within the physics engine.

VI. CONCLUSIONS

In this work, we introduce Placeit!, a framework for the automatic generation of rigid object placement poses in simulation. Our method is versatile, applying to various scenarios like table-top, hanging, stacking, and insertion. We demonstrate that a Placeit!-based approach to pick-and-place achieves a 90% success rate, proving the framework's efficiency in teaching robots manipulation skills for known objects in open environments. Furthermore, the high quality of the generated synthetic data suggests that Placeit! is a highly promising tool for building future foundation models in robotics.

ACKNOWLEDGMENT

This work was supported by the German Ministry of Education and Research (BMBF; grant 01IS21080), the French Agence Nationale de la Recherche (Learn2Grasp, ANR-21-FAI1-0004; Tactile, ANR-25-CE33-2325), and the European Union's Horizon Europe Framework Programme (grant agreements No. 101070381 and No. 101070596).

REFERENCES

[1] Zhang, H., Wu, Z., Huang, L., Christen, S., & Song, J. (2025). RobustDexGrasp: Robust Dexterous Grasping of General Objects. arXiv.

[2] Padalkar, A., Pooley, A., Jain, A., Bewley, A., Herzog, A., Irpan, A., ... & Jain, V. (2023). Open x-embodiment: Robotic learning datasets and rt-x models. arXiv.

[3] Zhao, Y., Bogdanovic, M., Luo, C., Tohme, S., Darvish, K., Aspuru-Guzik, A., ... & Garg, A. (2025). AnyPlace: Learning Generalized Object Placement for Robot Manipulation. arXiv.

[4] Jermann, T., Kolvenbach, H., Estay, F. E., Kramer, K., & Hutter, M. (2024). An Efficient Multi-Robot Arm Coordination Strategy for Pick-and-Place Tasks using Reinforcement Learning. arXiv.

[5] Cao, H., Fang, H. S., Liu, W., & Lu, C. (2021). Suctionnet-1billion: A large-scale benchmark for suction grasping. RAL.

[6] Huber, J., Hélénon, F., Coninx, M., Ben Amar, F., Doncieux, S. (2023). Quality Diversity under Sparse Reward and Sparse Interaction: Application to Grasping in Robotics. ECJ.

[7] Eppner, C., Mousavian, A., Fox, D. (2021, May). Acronym: A large-scale grasp dataset based on simulation. ICRA 2021.

[8] Noh, S., Kang, R., Kim, T., Back, S., Bak, S., & Lee, K. (2024). Learning to place unseen objects stably using a large-scale simulation. RAL.

[9] Simeonov, A., Goyal, A., Manuelli, L., Yen-Chen, L., Sarmiento, A., Rodriguez, A., ... & Fox, D. (2023). Shelving, stacking, hanging: Relational pose diffusion for multi-modal rearrangement. arXiv.

[10] Firoozi, R., Tucker, J., Tian, S., Majumdar, A., Sun, J., Liu, W., ... & Schwager, M. (2025). Foundation models in robotics: Applications, challenges, and the future. IJRR.

[11] Kim, M. J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., ... & Finn, C. (2024). Openvla: An open-source vision-language-action model. arXiv.

[12] Khazatsky, A., Pertsch, K., Nair, S., Balakrishna, A., Dasari, S., Karamcheti, S., ... & Finn, C. (2024). Droid: A large-scale in-the-wild robot manipulation dataset. arXiv.

[13] Barreiros, J., Beaulieu, A., Bhat, A., Cory, R., Cousineau, E., Dai, H., ... & TRI LBM Team. (2025). A careful examination of large behavior models for multitask dexterous manipulation. arXiv.

[14] Walke, H. R., Black, K., Zhao, T. Z., Vuong, Q., Zheng, C., Hansen-Estruch, P., ... & Levine, S. (2023, December). Bridgedata v2: A dataset for robot learning at scale. CoRL.

[15] Lum, T. G. W., Matak, M., Makoviychuk, V., Handa, A., Allshire, A., Hermans, T., ... & Van Wyk, K. (2024). Dextrah-g: Pixels-to-action dexterous arm-hand grasping with geometric fabrics. arXiv.

[16] Su, E., Jia, C., Qin, Y., Zhou, W., Macaluso, A., Huang, B., & Wang, X. (2024, May). Sim2real manipulation on unknown objects with tactile-based reinforcement learning. ICRA 2024.

[17] Bjorck, J., Castañeda, F., Cherniadev, N., Da, X., Ding, R., Fan, L., ... & Zhu, Y. (2025). Gr00t n1: An open foundation model for generalist humanoid robots. arXiv.

[18] Patel, S., Yin, X., Huang, W., Garg, S., Nayyeri, H., Fei-Fei, L., ... & Li, Y. (2025). A real-to-sim-to-real approach to robotic manipulation with VLM-generated iterative keypoint rewards. arXiv.

[19] Hussing, M., Mendez, J. A., Singrodia, A., Kent, C., & Eaton, E. (2023). Robotic manipulation datasets for offline compositional reinforcement learning. arXiv.

[20] Huber, J., Hélénon, F., Watrelot, H., Amar, F. B., & Doncieux, S. (2024). Domain Randomization for Sim2real Transfer of Automatically Generated Grasping Datasets. ICRA 2024.

[21] Miller, Andrew T., and Peter K. Allen. "Graspit! a versatile simulator for robotic grasping." RAM.

[22] Cully, A., Mouret, J. B., & Doncieux, S. (2022, July). Quality-diversity optimisation. GECCO Companion 2022.

[23] Garrett, C. R., Lozano-Perez, T., & Kaelbling, L. P. (2018). Ffrob: Leveraging symbolic planning for efficient task and motion planning. IJRR.

[24] Wan, W., Igawa, H., Harada, K., Onda, H., Nagata, K., & Yamanobe, N. (2019). A regrasp planning component for object reorientation. Autonomous Robots.

[25] Li, L., Yang, G., Shao, L., & Hsu, D. (2024). Stable Object Placement Under Geometric Uncertainty via Differentiable Contact Dynamics. arXiv.

[26] Simeonov, A., Goyal, A., Manuelli, L., Yen-Chen, L., Sarmiento, A., Rodriguez, A., ... & Fox, D. (2023). Shelving, stacking, hanging: Relational pose diffusion for multi-modal rearrangement. arXiv.

[27] Shan, D., Mo, K., Yang, W., Chao, Y. W., Fouhey, D., Fox, D., & Mousavian, A. (2025). Slot-Level Robotic Placement via Visual Imitation from Single Human Video. arXiv.

[28] Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., ... & Zaremba, W. (2017). Hindsight experience replay. NeurIPS.

[29] Fang, H. S., Wang, C., Fang, H., Gou, M., Liu, J., Yan, H., ... & Lu, C. (2023). Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. T-RO.
[30] Lertkultanon, P., & Pham, Q. C. (2018). A certified-complete bimanual manipulation planner. T-ASE.
[31] Harada, K., Tsuji, T., Nagata, K., Yamanobe, N., & Onda, H. (2014). Validating an object placement planner for robotic pick-and-place tasks. RAS.
[32] Street, C., Lacerda, B., Mühlig, M., & Hawes, N. (2024). Right place, right time: Proactive multi-robot task allocation under spatiotemporal uncertainty. JAIR.
[33] Macé, V., Boige, R., Chalumeau, F., Pierrot, T., Richard, G., & Perrin-Gilbert, N. (2023, July). The quality-diversity transformer: Generating behavior-conditioned trajectories with decision transformers. GECCO 2023.
[34] Huber, J., Hélénon, F., Kappel, M., Chelly, E., Khoramshahi, M., Amar, F. B., & Doncieux, S. (2024, October). Speeding up 6-DoF Grasp Sampling with Quality-Diversity. IROS 2024.
[35] Morrison, D., Corke, P., & Leitner, J. (2020). Egad! an evolved grasping analysis dataset for diversity and reproducibility in robotic manipulation. RAL.
[36] Mouret, J. B., & Clune, J. (2015). Illuminating search spaces by mapping elites. arXiv.
[37] Haustein, J. A., Hang, K., Stork, J., & Kragic, D. (2019, November). Object placement planning and optimization for robot manipulators. IROS 2019.
[38] Batlle, J. A., & Condomines, A. B. (2022). Rigid body dynamics. Cambridge University Press.
[39] Fontaine, M., & Nikolaidis, S. (2023, July). Covariance matrix adaptation map-annealing. GECCO 2023.
[40] Eppner, C., Mousavian, A., & Fox, D. (2019, October). A billion ways to grasp: An evaluation of grasp sampling schemes on a dense, physics-based grasp data set. ISRR 2019.
[41] Calli, B., Walsman, A., Singh, A., Srinivasa, S., Abbeel, P., Dollar, A. M. (2015). Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols. arXiv.
[42] Pavlichenko, D., & Behnke, S. (2025). Dexterous Pre-grasp Manipulation for Human-like Functional Categorical Grasping: Deep Reinforcement Learning and Grasp Representations. T-ASE.
[43] Hampali, S., Hodan, T., Tran, L., Ma, L., Keskin, C., Lepetit, V. (2023). In-hand 3d object scanning from an rgb sequence. CVPR.
[44] Hélénon, F., Huber, J., Amar, F. B., Doncieux, S. (2023). Learning to Grasp: from Somewhere to Anywhere. ICRA 2024 MoMa workshop.

Supplementary Materials

APPENDIX I
EXPERIMENTAL DETAILS: SIMULATION

Algorithms. Let μ be the population size, λ the number of offspring, k the number of neighbors considered for novelty computation, and N_e the maximum number of evaluation. We set: $\mu = \lambda = 500$, $k = 15$, $N_e = 100$k. All offspring are mutated with a probability $ind_{pb} = 0.3$ to modify each gene. The mutation operator applied by default to all the methods is a Gaussian perturbation of 0 mean and 0.1 standard deviation. For *CMA_MAE* variants, the same parameters as in Fontaine et al. were used [39]: The emitter batch size is set to 36, and the number of emitters to 15 and $\alpha = 0.2$. Finally, we set $f_{min} = -10000$ to make sure $f_i > f_{min}$).

Heatmaps. The heatmaps in Fig. 6 were computed based on a *relative quality*: for a given archive of successful pose, we extract a chunk of object point cloud of 10% along the z axis from the table surface. Each chunk is an approximation of the points on the object surface that are in contact with the table. The obtained points are associated with the computed fitness for each chunk. All of the collected fitnesses for each points are then summed. The relative quality for a given point is computed as the ratio between its associated sum of fitnesses and the maximum value among all the sums. This allows to obtain a score between 0 and 1, where scores close to 1 emphasize points that are often in contact with the table and associated with a high value of fitness. This computation is similar to the QD-score, which is commonly used to encapsulate both the diversity and the quality in a single value [22].

APPENDIX II
EXPERIMENTAL DETAILS: PHYSICAL WORLD

Data generation. To compute $\sigma_i^{o^{DR}}$, an axis of perturbation (among the translation and the rotation ones) is first randomly sampled. The magnitude of the perturbation depends on the mass of the object, such that the perturbations on each object are comparable.

All the remaining hyperparameters are similar to those used in the simulated experiments.

Real world setup. The gripper was controlled with a joint impedance controller with gravity compensation. No additional material that could increase adhesion is used to make grasping easier. The gravity compensation is computed with respect to the fixed arm base. We used a custom vision code derived from [44] for getting the position and orientations of the objects o and s. The motion planning of the end effector is conducted with RRT connect through *Moveit!*, assuming that the object and the table are collision bodies.